四足歩行ロボットの設計と実装
―災害時の情報収集用ロボットのプロトタイピング―

野　津　伸　治[1]

Shinji Notsu：A Design and Implementation of a Quadruped Robot

本研究の目的は，災害時に情報収集するロボットの量産を想定した場合のプロトタイプの設計と実装である．四足歩行を行うためのフレームの設計と歩行アルゴリズムの検証に焦点をあてたものである．成果としては，第１にはサーボブラケットの加工のしやすさと精度と強度を考慮してアクリル板のＬ字アングルを利用したこと．第２には，片軸サーボモーターの回転精度を安定させるための両軸化を行ったこと．第３には，前後左右の移動速度の一定化とアルゴリズム単純化のためにボディの対称設計を行ったこと．第４として，四足を８自由度で実装する場合に，サーボモーターの回転範囲を最大限確保できる配置を提案したこと．第５として，前後左右の四足歩行アルゴリズムの実装ができたことである．

キーワード：四足歩行ロボット　マイコン制御　静的歩行

はじめに

ロボットは，日本の社会において，産業界では工場における製造ラインの組み立て用途や，警備用途など様々な分野で長い間用いられてきた．また一般家庭向けでも，愛玩用や癒し用途，清掃用途など様々な製品が浸透している．また近年の異常気象等によって様々な災害が日本各地で発生し，人命救助や状況把握には情報収集が必要不可欠である．さらに大型公共建築物の耐用年数の経過に伴う現状把握の情報収集も必要不可欠である．人間が現場で活動するには放射能の影響や建物の崩壊の恐れがある，足場を組んで大掛かりな準備をしないと実行できないなど，必要性はあるが実行がとても難しい場合に，各種センサーと通信機能を持ったロボットによる代行はとても有意義である．災害時の情報収集現場を考えると，置かれている移動条件の厳しさから車輪を使うメカニズムによる移動方法は制約が多いといえる．そこで移動自由度の高さと姿勢維持の安定度から四足歩行ロボットによる実装が有益であると考える．

1　鳥取短期大学生活学科

１．ロボットの基本設計

災害時情報収集ロボットのプロトタイプを実装するにあたって，歩行の安定度とメカニズムの簡素化から四足歩行で８自由度のロボットの設計を目標とする．１本の足に２自由度で３次元空間（側面から見て前後に回転するピッチ軸，正面から見て左右に回転するロール軸，上から見て地面と平行に回転するヨー軸）では２平面の自由度が確保されることになる．ロボットは前後左右に移動が可能であり，移動する平面は必ずしもフラットな面ではなく，ある程度の凹凸のある所も移動できるものとする．また人が入れない空間にも入れるようなサイズを想定する．

量産も想定してプロトタイプは試行錯誤を繰り返すためにフレーム加工は簡単に行えて，特殊な工具

は利用しないものとする.

2．ロボットの電子回路

基本的な構成は複数のサーボモーターをマイクロコンピュータから制御するものとする．様々な歩行のパターンをプログラムで変更できるものとする．将来の足の自由度の増加にも対応できるものとする. 従って, マイクロコンピュータとサーボモーターは直接接続するのではなく，中間に制御ボードを介在して行う.

（1） マイクロコンピュータ

プロトタイプでは最大 8 個のサーボモーターを制御するために 8bit マイクロコンピュータ ATmega328P を搭載するオープン・ハードウェアのマイコンボードの Arduino nano 互換機を用いた．マイクロコンピュータと制御ボードの間で I^2C 通信を行える機能を有している．プログラミングするときに利用できる processing 言語のライブの充実もマイコン選択の決め手である.

（2） サーボモーター

プロトタイプの開発において本体重量の軽量化を最大限に図るためとコスト削減の観点からアナログサーボモーターSG90 互換機を利用した.

表 1 は SG90 の物理的及び電気特性の仕様である.

表 1．SG90 サーボモーターの仕様

PWMサイクル	20mS
制御パルス	0．5ms～2．4ms
制御角	±約90°（180°）
配線	茶＝GND、赤＝電源［+］、橙＝制御信号　［JRタイプ］
トルク	1．8kgf・cm
動作速度	0．1秒／60度
動作電圧	4．8V（～5V）
温度範囲	0℃～55℃
外形寸法	22．2x11．8x31mm
重量	9g

SG90 サーボモーターのブラケットを自作するためにこの形状の寸法の詳細をノギスで計測してまとめたものが図 1 のとおりである.

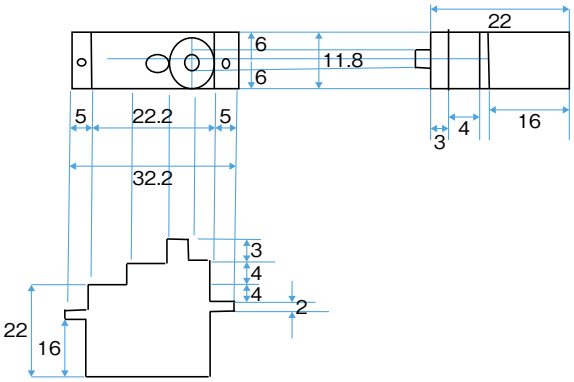

図 1．SG90 サーボモーターの詳細な寸法(単位：mm)

試作の段階で片軸サーボモーターの動作の精度の問題が生じたので，SG90 を分解して，軸の反対側のフレームにドリルで穴を開けて，半径Φ =2mm で長さ L=10mm のネジを通して両軸化することで回転精度の著しい向上が確認できたのでこれを生かすブラケットを製作することにした.

（3） サーボ制御コントローラー

1 個の制御コントローラーで最大 16 個のサーボモーターを 12bit の分解能で扱うことができれば十分であるので PCA9685 を用いることにする.

Arduino nano と PCA9685 の結線は表 2 のとおりである.

表 2. マイコンボードと PCA9685 の結線

AE-PCA9685 側	ArduinoUNO 側 (例)
+5V	未接続 (ブロック端子よりサーボ電源供給)
OE	未接続
SCL	A5
SDA	A4
GND	GND
VCC	5V

（4） 回路の実態配線図

マイコンボード Arduino nano にマイコン制御ボード PCA9685 を経由して配線した．サーボモーターSG90 を 8 個制御するマイコンボード用とサーボモーター用のそれぞれの電源も含めた全体の回路図を図 2 に示す.

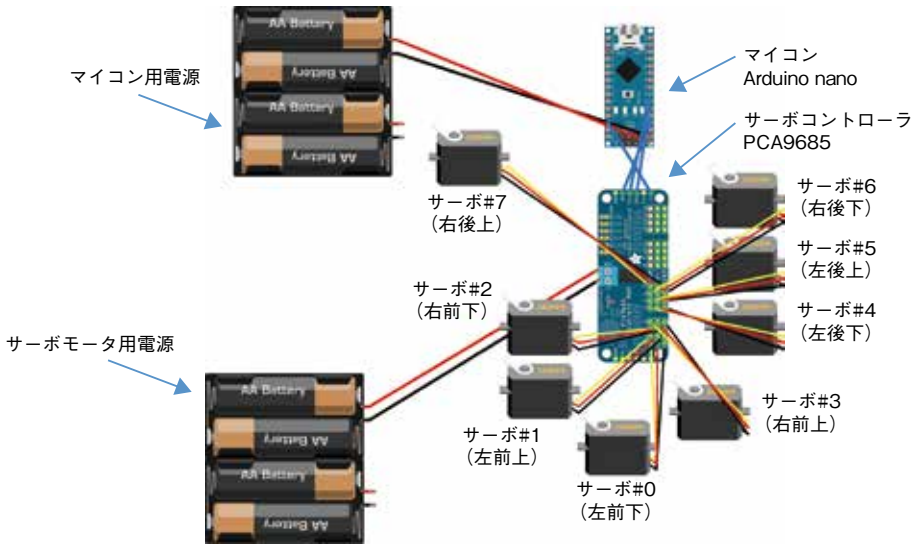

図 2. 四足歩行ロボットの電子回路実態配線図

3．ロボットのフレーム設計

四足歩行を実現するためには，例えば前進なら前後左右の足のうち，まず右前足を一歩前に，次に左後ろ足を一歩前に，続いて左前足を一歩前に，最後に右後ろ足を一歩前に動かすことを繰り返す必要がある．これは 3 次元空間での最小限 1 空間（ピッチ軸）の移動で実現できる．フレーム加工と想定した歩行アルゴリズムの試行の推移を以下で説明する．

（1） スチレンボードによる 2 次元フレーム

サーボフレーム自体は 3 次元である必要があり，強度の確保からアルミ板を加工して製作される場合が多い．しかしプロトタイプの製作であるので歩行アルゴリズムを考慮し，加工のしやすさを優先して建築模型などで用いられるスチレンボードを利用して加工を行った．サーボブラケットとしてのスチレンボードの切り出しは図 3 のように行った．

実際にスチレンボードから切り出してフレームとしてサーボモーターを組み込んだものが図 4 である．

カッターナイフでの切り出しはとても簡単であったが，加工の精度と強度の観点で安定した歩行の実現が難しかった．

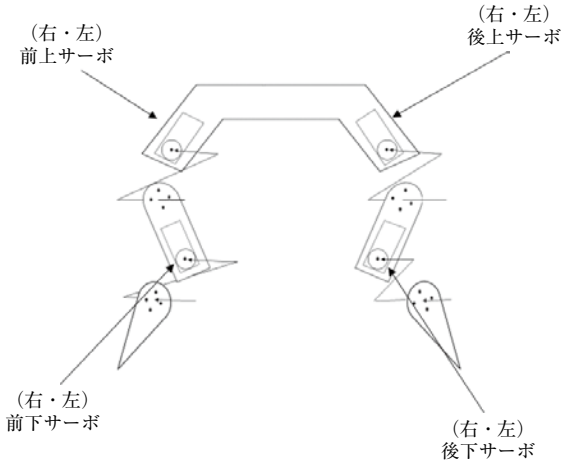

図 3. ロボットの 2 次元フレーム

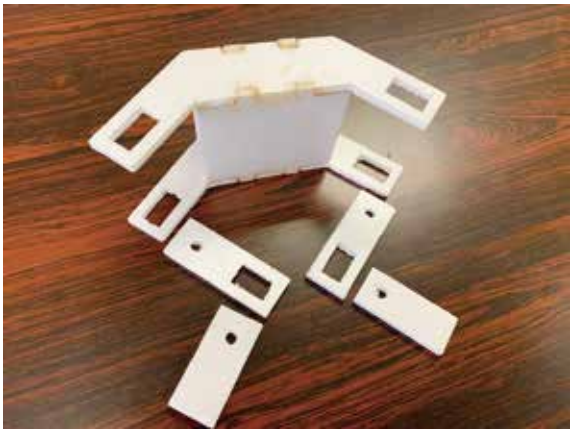

図 4. 切り出したスチレンボードでのフレーム

（2） L 字アクリル板による 3 次元フレーム A

個々の足の構造の改善と形成用の素材の見直しが課題であった．素材をスチレンボードからアクリル板に変更することで，重量は増すが強度が確保でき

る．また平面を垂直に貼り合わせるもしくは融着する場合の精度が問題になると考え，L 字のアングル状態のアクリル板を切り出して加工することで問題の解決を図った．更なる加工上の問題は，サーボブラケットを組み合わせる場合に，ブラケットの形状が複雑になり，組み立ての難易度が上がることである．また加工すべき形状のパターンが格段に増えた．

2 自由度の足の先端を座標（x，y）へ移動するために 2 つのサーボモーターに与える角度θ_1とθ_2の関係を考察する．前提として上側の足の長さをL_1，下側の足の長さをL_2とする．L_1を座標（u，v）へ移動するために角度θ_1動かし，さらにL_2を座標（x，y）へ移動するために角度θ_2動かす．これを図示したものが図 5 である．（式 1）は角度θ_1とθ_2，および長さL_1とL_2から座標（x，y）を表す．

$$\begin{cases} x=L_1\cos\theta_1+L_2\cos(\theta_1+\theta_2) \\ y=L_1\sin\theta_1+L_2\sin(\theta_1+\theta_2) \end{cases} \quad \sim(式1)$$

～角度θ_1とθ_2を入力→アーム先端の座標（x, y）
L_1，L_2…二つのアームの長さ
θ_1，θ_2…二つの関節の角度
(0, 0)…根元の関節の位置
(x, y)…アームの先端部分の座標

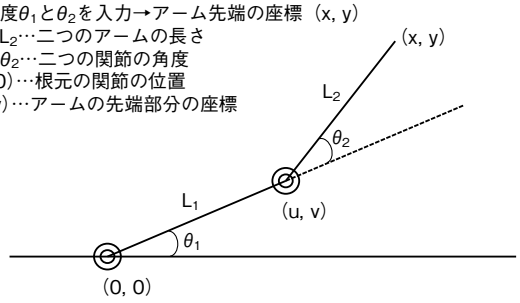

図 5．順運動：2 自由度の足の長さ，角度と位置の関係

マイコンから各サーボモーターへ与える命令は角度θ_1とθ_2になるので，図 5 の逆関数を求めなければならない．順運動（式 1）の逆運動に相当する（式 2）と（式 3）である．

$$\theta_2=\pm\mathrm{atan2}\sqrt{((L_1+L_2)^2-(x^2+y^2))/((x^2+y^2)-(L_1-L_2)^2)} \quad \sim（式2）$$

$$\begin{cases} \theta_1=\mathrm{atan2}(y/x)-\mathrm{atan2}(L_2\sin\theta_2)/(L_1+L_2\sin\theta_2) \\ \theta_1=\mathrm{atan2}(y/x)+\mathrm{atan2}(L_2\sin\theta_2)/(L_1+L_2\sin\theta_2) \end{cases} \sim(式3)$$

これらの式にアーム先端の座標（x，y）を入力すると関節θ_1とθ_2の角度が出力される．

これに基づき 2 自由度の 1 本の足の動かし方を図示したものが図 6 である．

この原理を考慮してサーボブラケットの加工とフレームとしての組立を考慮したものが図 7 である．

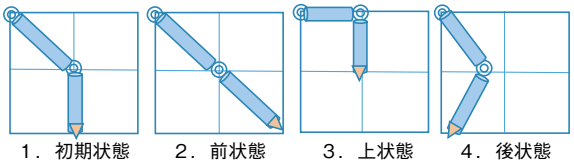

図 6．2 自由度の 1 本足の運び

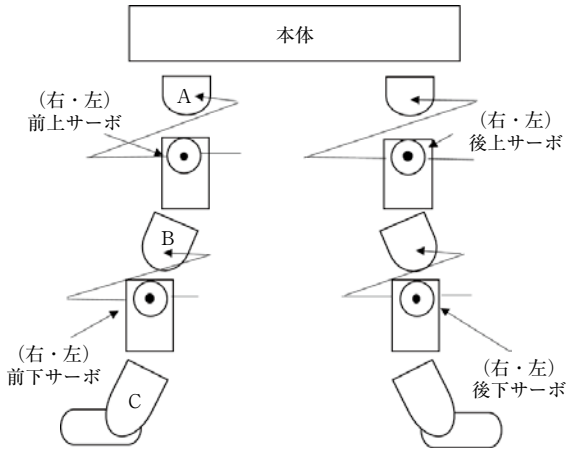

図 7．前後の足の配置

この形状で実装した場合，前後方向の移動では自然な歩行が実現できたが，左右方向の移動はぎこちなく時間も前後方向の移動の数倍かかる．

（3） L 字アクリル板による 3 次元フレーム B

3 次元空間で 1 平面（ピッチ軸）の移動精度にサーボモーター2 個を利用するのではなく，2 平面（ピッチ軸とヨー軸）に 1 サーボモーターずつを用いることで前後左右の移動の精度と速度を確保する形に歩

表 3．第 3 版のロボットのパーツ一覧

部品一覧	仕様	個数	用途
ネジ	φ=2㎜/L=16㎜	12	サーボの軸
ネジ	φ=2㎜/L=10㎜	12	サーボブラケット
ネジ	φ=2㎜/L=8㎜	16	サーボブラケット
ネジ	φ=2㎜/L=6㎜	4	サーボブラケット
ネジ	φ=2㎜/L=4㎜	4	サーボブラケット
ビス	φ=2㎜	30	サーボブラケット
ワッシャー	φ=2㎜	30	サーボブラケット
タッピング	φ=2㎜/L=4㎜	16	サーボホーン固定
アクリル板	100×100㎜、厚さ=2㎜	1	ボディ
アングル（プラスチック）	40×40×1600㎜、厚さ2㎜	1	サーボブラケット
角材（プラスチック）	5×5×200㎜	1	足の下側
マイコンArduino	Arduino nano v3.3	1	CH340
USBケーブル	miniUSB(B)-USB(A)	1	
マイクロサーボ	SG90	8	橙＝PWM/赤＝V＋/茶＝GND
サーボコントローラ	PCA9685	1	16CH,12bit,Adafruits Servo Drivers
電池ボックス（スイッチ・リード線）	単4乾電池×4本	2	マイコン用とサーボ用
アルカリ乾電池	単4乾電池	8	マイコン用とサーボ用
ジャンパーワイヤー	オスーオス	10	
ジャンパーワイヤー	オスーメス	5	
ラバー			足先の滑り防止

行アルゴリズムと対応するサーボブラケットの接合の仕方を変更した．表3がこの設計に基づく利用パーツの一覧である．

アクリル製のＬ字アングルからサーボブラケットを切り出して加工する際の寸法と穴の位置は図8のように行った．

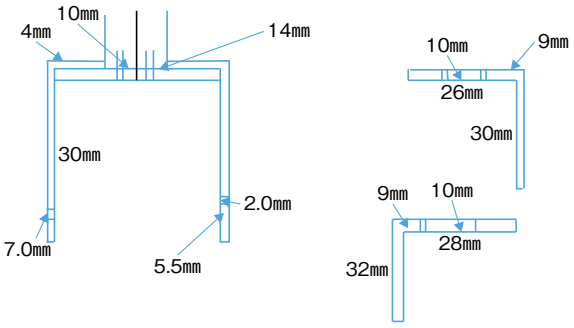

図8．Ｌ字アングルからサーボブラケットの加工

図9は図8で加工したブラケットにサーボモーターSG90を組み込み回転させるイメージである．

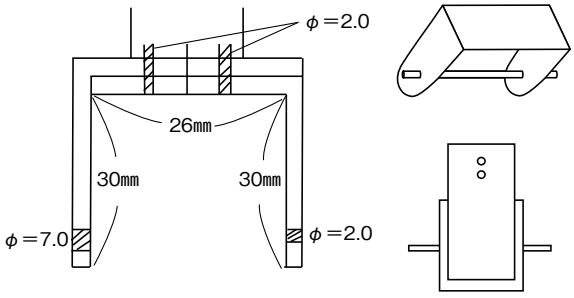

図9．サーボモーターの回転

図10はボディに四本の足を取り付ける位置を示している．

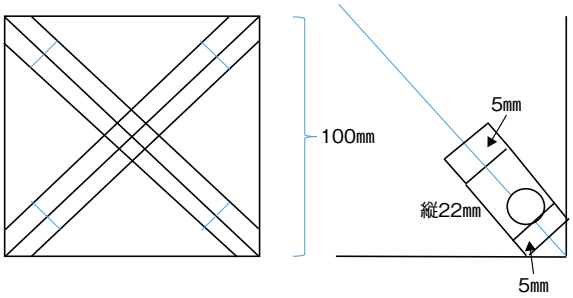

図10．ボディへの四本足の取付け位置

図11はボディとサーボブラケットの位置を示している．

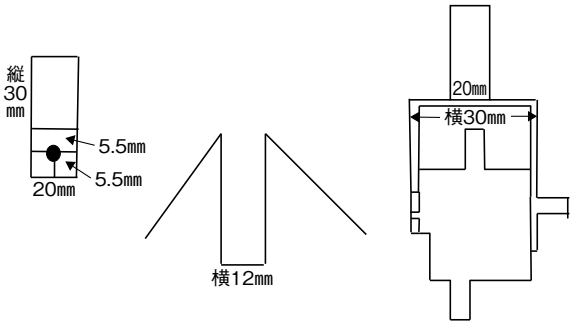

図11．ボディへのサーボブラケットの位置

図12はボディとブラケットの組み合わせを示している．

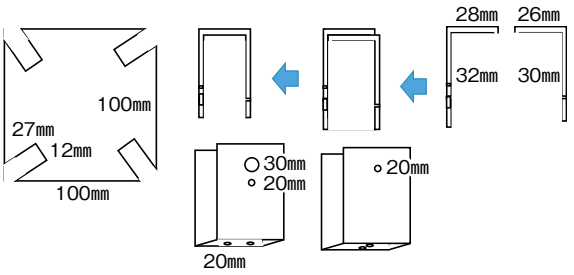

図12．ボディとブラケットの組み合わせ

図13はブラケットを直交させる組合せと足の下側の加工イメージである，

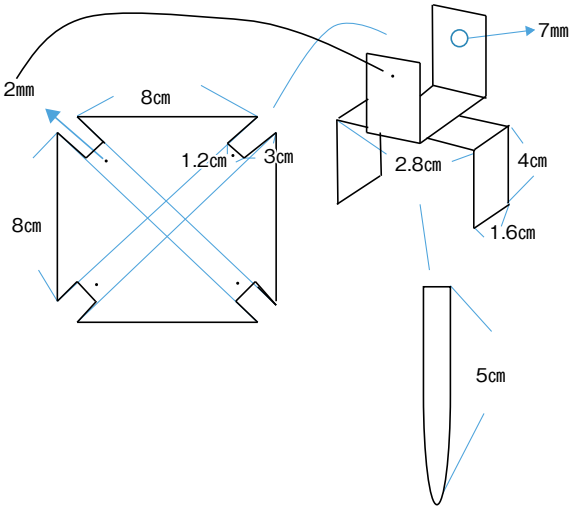

図13．ブラケットの直交と足の下側

以上の過程を経て組み上げた四足歩行ロボットの全景が図14と図15である．

当初設計通りに前後左右の動作の精度と速度が確保できた．

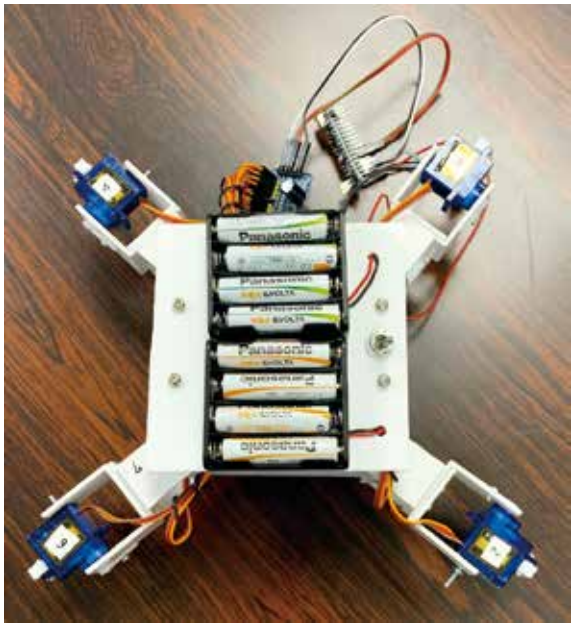

図 14.　四足歩行ロボットの全景（真上から）

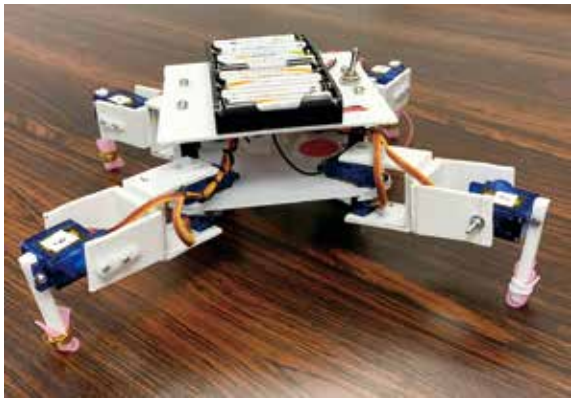

図 15.　四足歩行ロボットの全景（斜め上から）

4．ロボットの歩行アルゴリズム

四本足で歩行する動物が前進する場合の足の運びを観察すると，例えば，まず右前足を前に動かし，次に左後ろ足を前に動かし，続いて左前足を前に動かして，最後に右後ろ足を前へ動かすことの繰り返しを行っていることがわかる．動物は敢えて安定状態を乱すことで自然な動きをしており，このことは動的歩行と呼ばれている．一方同じ四本足で動くロボットの動作はぎこちない．これは，四本足のうち地面に接している足（接地足）三本で定義される 3 角形の内側に重心が常にあるように非接地足を動かしているからである．この歩行は静的歩行（図 16）と呼ばれている．

今回はプロトタイプの開発のため後者の静的歩行で実装していくことにする．

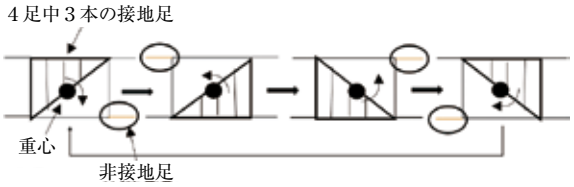

図 16.　静的歩行

（1）　アルゴリズム

四足歩行ロボットの 4 つの足の上部と下部にそれぞれ割り当ててあるサーボモーターに，0 番から 7 番までの番号を割り当てて制御の時の識別子にする（図 17）．

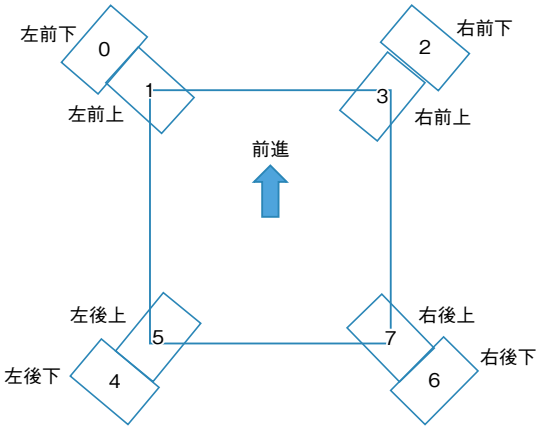

図 17.　四本 8 自由度の足とサーボモーターの対応番号

図 18 に，静的歩行の足の運びをロボットの真上から見た図（ヨー軸）と，その時にサーボモーターが動作させるべき角度を決定していくことにする．黒色の太い矢印は接地足を表し，赤色の太い矢印がその時動かす足を表すこととする．但し赤い矢印は，非接地足として移動する場合と接地足としてボディを前へ押し出す場合とがある．

図 18 の①は，ロボットが通電されたときのホームポジション（稼働角度の範囲は 0 度～180 度のうち，すべてのサーボモーターが 90 度を示している）を表している．図 18 の②は前進するときの開始状態を表している．ステップ I として①から②の状態へ遷移するために右前足を後ろへ，右後ろ足は前へ

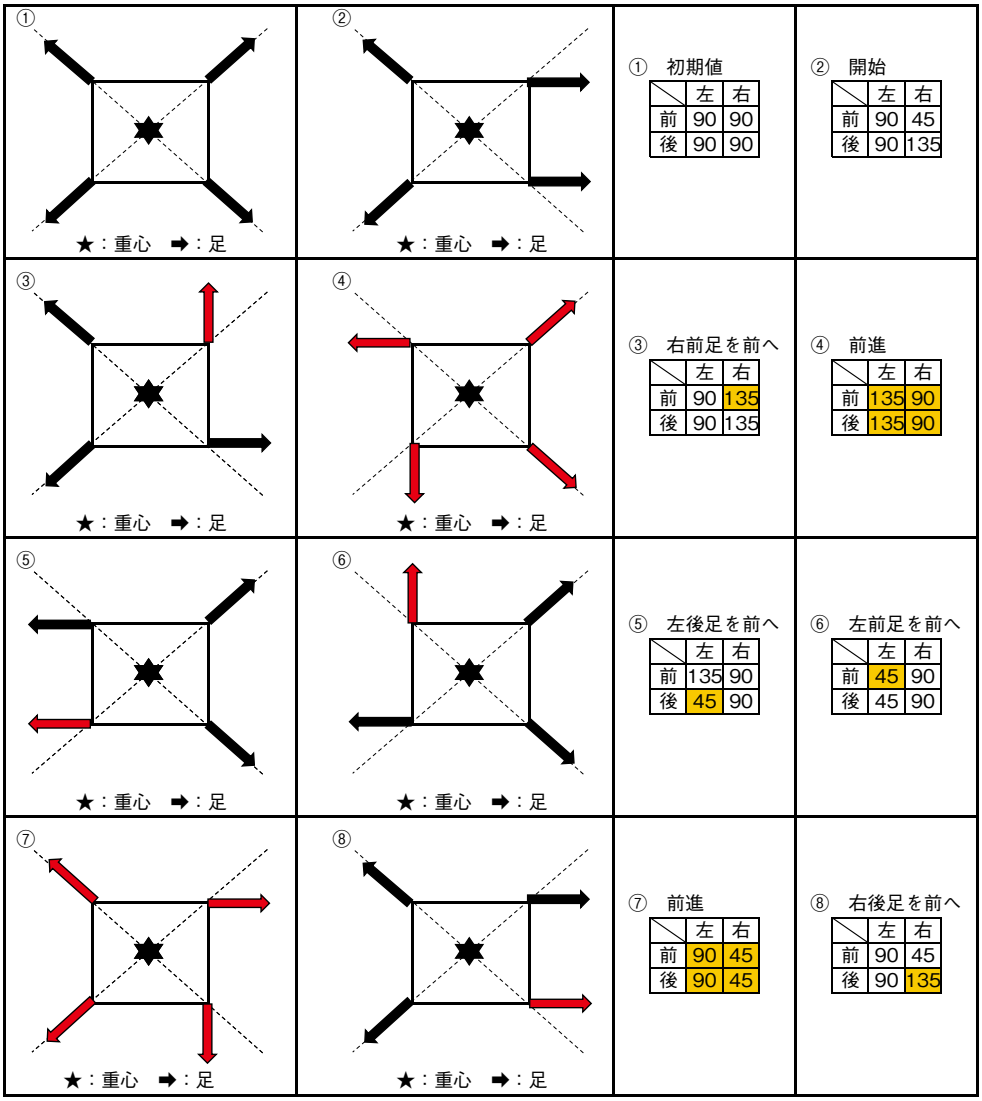

図 18. 静的歩行の四本の足の動作と角度

動かしている．図 18 の左側が足の運びの図示で，右側がサーボモーターの角度を表している．次のステップⅡ（②から③への遷移）として，右前足を一旦浮かしてから前へ動かしている．ステップⅢ（③から④への遷移）で四本の足すべてを接地したまま同時に後ろに動かすことでボディを前へ動かすことができる．ステップⅣ（④から⑤への遷移）で，左後ろ足を前へ動かしている．この状態⑤は状態②の線対称になっている．ステップⅤ（⑤から⑥への遷移）で左前足を一旦浮かしてから前へ動かしている．ステップⅥ（⑥から⑦への遷移）で，四本の足すべてを接地したまま同時に後ろへ動かすことでボディを前進させている．ステップⅦ（⑦から⑧への遷移）で，右後ろ足を一旦浮かしてから前へ動かしている．その結果の⑧の状態はまさに②の状態であり，これを繰り返すことでボディを前進させ続けていくことが可能である．

ロボットの後退や左右への移動は，ボディを対称に実装したので，動く方向に合わせてサーボモーターの役割を読み替えることで実現できる．

この一連の処理の流れを流れ図で表すと図 19 のようになる．

図 19 は歩行の基本アルゴリズムであるので，対象とするマイコンボードやサーボモーターをより高性能なものに交換して量産タイプで実装する場合も，このアルゴリズムを適用することができる．

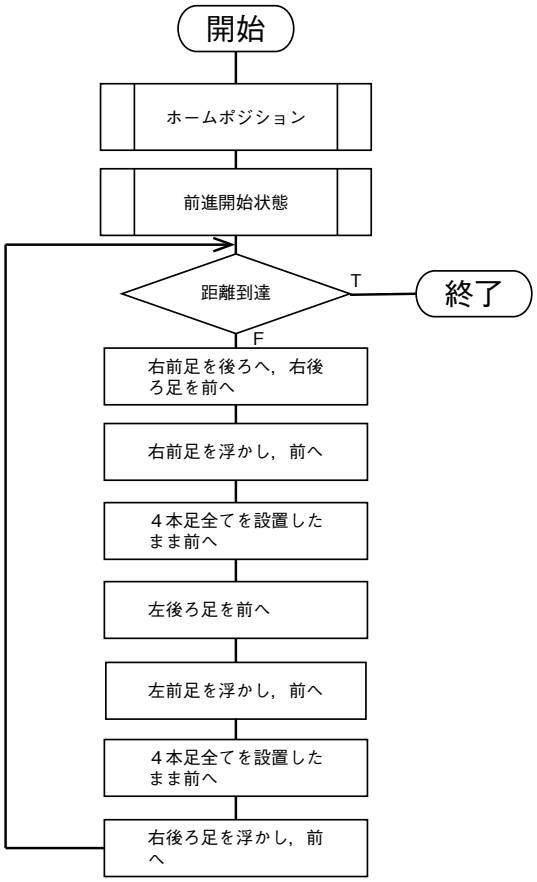

図 19. 前進の流れ図

（2） Processing 言語による実装

Arduino nano のプログラム開発環境は Arduino IDE[1),2)] で，最新の version 1.8.9 を利用した．利用したプログラミング言語は Processing である．

Processing 言語向けに PCA9685 用サーボモーター・ライブラリー[3)]が提供されているので利用した．

また，各サーボモーターの位置出しを行うことで，サーボモーターの初期状態の角度を定義しておくことがその後の歩行の出発点となる．

更に注意が必要な点は，サーボブラケットの組み合わせによって位置出しされた角度からプラスマイナスの移動量が逆転する場合があるので，フレームとして組立後，四本の足の上下で記録しておくことである．

図 18 の範囲で 8 個のサーボモーターが動作することを確認した Processing でのコードが図 20 である．

以上を踏まえて当初の設計通りの前後左右の移動が行える四足歩行ロボットの稼働部に特化した実装を確認することができた．

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
#define SERVOMIN 150 // this is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 600 // this is the 'maximum' pulse length count (out of 4096)
void setup(){
  pwm.begin();
  pwm.setPWMFreq(60); // Analog servos run at ~60 Hz updates
  servo_write(0,90);
}
void servo_write(int n, int ang){ // 動かすサーボと角度を引数に持つ
  ang = map(ang, 0, 180, SERVOMIN, SERVOMAX); // 角度（0～180)をPWMのパルス幅(150～600)へ変換
  pwm.setPWM(n, 0, ang);
}
void loop(){
 //=======<<前進 >>============
 //-----<<右前足>>-----
  // 2番ピンのサーボに角度45度を指示
  servo_write(2,45);
  delay(500);
  // 3番ピンのサーボに角度135度を指示
  servo_write(3,135);
  delay(500);
  // 2番ピンのサーボに角度90度を指示
  servo_write(2,90);
  delay(500);
  // 3番ピンのサーボに角度90度を指示
  servo_write(3,90);
  delay(500);

 //-----<<左後足>>-----
  // 4番ピンのサーボに角度45度を指示
 servo_write(4,45);
 delay(500);
 // 5番ピンのサーボに角度45度を指示
 servo_write(5,45);
 delay(500);
 // 4番ピンのサーボに角度90度を指示
 servo_write(4,90);
 delay(500);
 // 5番ピンのサーボに角度90度を指示
 servo_write(5,90);
 delay(500);

 //-----<<左前足>>-----
  // 0番ピンのサーボに角度45度を指示
 servo_write(0,45);
 delay(500);
 // 1番ピンのサーボに角度45度を指示
 servo_write(1,45);
 delay(500);
 // 0番ピンのサーボに角度90度を指示
 servo_write(0,90);
 delay(500);
 // 1番ピンのサーボに角度90度を指示
 servo_write(1,90);
 delay(500);

 //-----<<右後足>>-----
 // 6番ピンのサーボに角度45度を指示
 servo_write(6,45);
 delay(500);
 // 7番ピンのサーボに角度135度を指示
 servo_write(7,135);
 delay(500);
 // 6番ピンのサーボに角度90度を指示
 servo_write(6,90);
 delay(500);
 // 7番ピンのサーボに角度90度を指示
 servo_write(7,90);
 delay(500);

}
```

図 20. 四本足 8 自由度の動作確認のための Processing でのコード

おわりに

前後左右に移動する 8 自由度四足歩行ロボットの躯体と制御アルゴリズムの設計と実装ができた．本研究の成果としては，第 1 にはサーボブラケットの加工のしやすさと精度と強度を考慮してアクリル板のＬ字アングルを利用したこと．第 2 には，片軸サーボモーターの回転精度を安定させるための両軸化を行ったこと．第 3 には，前後左右の移動速度の一定化とアルゴリズム単純化のためにボディの対称設計を行ったこと．第 4 として，四足を 8 自由度で実装する場合に，サーボモーターの回転範囲を最大限確保できる配置を提案したこと．第 5 として，前後左右の四足歩行アルゴリズムの実装ができたことである．一連の設計と実装は鳥取短期大学生活学科情報・経営専攻 1 年生のグループワーク（1 グループ 5 名で 10 グループ）による PBL（課題解決型学習）の演習（8 回分）としても確認できた．

量産型に向けては，ボディの精緻化のため 3D プリンターによるブラケットの小型・軽量化が考えられる．また動作速度とトルクの向上に向けてデジタル・サーボモーターの利用が考えられる．

また災害時の情報収集に向けては，ジャイロや CO_2，温度，赤外線センサー，CCD カメラなどの各種センサーの搭載と，遠隔で操作するために IEEE802.11ac などの通信機能の搭載が必要不可欠である．

想定外の状況に応じた移動の方法としては，歩行アルゴリズムの自律的獲得が機械学習などで可能である．

引用・参考文献

1）Arduino IDE 統合開発環境
https://www.arduino.cc/en/Main/Software
2）CH340 ドライバ
http://www.wch.cn/download/CH341SER_ZIP.html
3）Arduino 用 Servo 制御ライブラリ
https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library